12/05/16 Assembly

Class Notes

Single instruction multiple data (cont.)

Final topics:

* Numbers & conversions
* Expressions
* Functions
* Arrays
* Hardware gpio
* SIMD

SIMD

* Uses a single line to perform functions for multiple data:
  + Ex) .data

w: .word 1,2,3,4

x: .word 5,6,7,8

y: .word 0,0,0,0

main:

//Load in first array

ldr r0, =w

vld1.32 {q0}, [r0]

//Load in second array

ldr r0, =x

vld1.32 {q1}, [r0]

//.s32 signifies that the values are a signed 32 bit value

//Multiply values

vmul.s32 q2, q0, q1

//Store values in array

ldr r0, =y

vst1.32 {q2}, [r0]

* Using multiple q registers
  + Items instantiated in data are continuous in memory
  + Ex) .data

w: .word 1,2,3,4

x: .word 5,6,7,8

y: .word 0,0,0,0

z: .word 0,0,0,0

main:

ldr r0, =w

ldr r1, =y

vld1.32 {q0,q1} [r0]

//Since there are only four values in array w[], this line will pull the next //32 bit values in memory (array x[])

* Strides in memory
  + When using vld#.32, you can specify the # to set the increments by which data is selected
    - Ex) vld2.32 //Strides of 2
      * Q0: 1,2,3,4
      * Q1: 5,6,7,8
      * Q2: 9,10,11,12
      * Q3: 13, 14, 15, 16
  + Make sure that your strides are consistent

GDB debugger

* Breakpoints
  + b main
    - creates a breakpoint at main
* Pulling register information
  + info register <register#>
* Pulling info from register
  + 